НЕЙРОІНФОРМАТИКА
ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

НЕЙРОИНФОРМАТИКА
И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

NEUROINFORMATICS
AND INTELLIGENT SYSTEMS

УДК 004.032.026

Bodyanskiy Ye.[1], Dolotov A.[2]

[1]Doctor of Science (Dr.-Ing. habil.), Kharkiv National University of Radio Electroniks
[2]Ph. D. student, Kharkiv National University of Radio Electroniks

ANALOG-DIGITAL SELF-LEARNING FUZZY SPIKING NEURAL NETWORK AND ITS LEARNING ALGORITHM BASED ON ‘WINNER-TAKES-MORE’ RULE

Analog-digital architecture of self-learning fuzzy spiking neural network is proposed in this paper. Spiking neuron synapse and some are treated in terms of classical automatic control theory. Conventional unsupervised learning algorithm of spiking neural network is improved by applying ‘Winner-Takes-More’ rule.

Key words: analog-digital architecture, self-learning fuzzy spiking neural network, automatic control theory, unsupervised learning algorithm, ‘winner-takes-more’ rule.

INTRODUCTION

Unsupervised classification, or clustering, is one of the fundamental problems of computational intelligence. Nowadays, there are various means for its solving, among them the significant place is occupied by artificial neural networks (self-organizing maps, ART neural networks, ‘Brain-State-in-a-Box’ neuromodels, etc.) [1], fuzzy clustering systems (fuzzy c-means, algorithms of Gustafson – Kessel, Yager – Filev, Klawonn – Hoeppner, etc) [2, 3], and hybrid systems based on the combination of both of them [4]. While the mentioned computational intelligence means for unsupervised data processing are well developed and rather powerful, they often appear to be inapplicable for real life problems solving due to their time-consuming run time. The present requires data processing systems to be not only powerful from computational point of view, but also to be sufficiently rapid to handle instantaneous changes in real world environment. New generation of artificial neural networks, commonly known as spiking neural networks [5, 6], challenged to overcome this drawback of the classical computational intelligence means. Being highly realistic models of biological neurons, spiking neural networks also inherited capability of rapid information processing from them [5, 7, 8]. Moreover, hybrid computational intelligence systems based on self-learning spiking neural network made it possible to extend capabilities of latter for efficient data clustering even under uncertainty [9–12]. Anyway, the highest efficiency of spiking neural networks can be achieved in the case of their hardware implementation only [8]. One can run into a difficulty, however, in this direction. Although spiking neural networks are becoming a popular computational intelligence tool for various technical problems solving that is confirmed by constant growing of the number of scientific and research works on that subject, their architecture and functioning are treated in terms of neurophysiology rather than in terms of any technical sciences apparatus. None technically plausible description of spiking neurons functioning has been proposed yet. Spiking neural network descriptions lack for general technical ground.

This work presents self-learning spiking neural network as a data processing system of classical control theory. Neuron synapse is shown to be a second-order critically damped response unit, and neuron soma is treated as a threshold-detection unit. Description of spiking neuron functioning in terms of the Laplace transform makes it possible to state spiking neural network architecture on a general technical ground that can be used in the following for constructing various hardware implementations of self-learning spiking neural networks.

SELF-LEARNING FUZZY SPIKING NEURAL NETWORK

Self-learning fuzzy spiking neural network architecture is heterogeneous three-layered feed-forward neural network with lateral connections in the second hidden layer [9].

The first hidden layer is constructed to perform population coding of input signal [7]. It acts in such a manner that each dimensional component $x_i(k)$, $i = \overline{1, n}$, of input signal $x(k)$ is processed by a pool of h receptive neurons RN_{li}, $l = \overline{1, h}$. Obviously, there can be different number of receptive neurons h_i in a pool for each dimensional component in the general case. For the sake of simplicity, we will consider here that the number of neurons is equal for all pools.

As a rule, activation functions of receptive neurons within a pool are bell-shaped (Gaussians usually), shifted, overlapped, of different width, and have dead zone. Generally firing time of a spike emitted by a receptive neuron RN_{li} upon incoming signal $x_i(k)$ lies in a certain interval $\{-1\} \cup [0, t_{\max}^{[0]}]$ referred to as coding interval and is described by the following expression:

$$t_{li}^{[0]}(x_i(k)) = \begin{cases} \lfloor t_{\max}^{[0]}(1 - \psi(|x_i(k) - c_{li}^{[0]}|, \sigma_{li})) \rfloor, \\ \qquad \psi(|x_i(k) - c_{li}^{[0]}|, \sigma_{li}) \geq \theta_{\text{r.n.}}; \\ -1, \ \psi(|x_i(k) - c_{li}^{[0]}|, \sigma_{li}) \geq \theta_{\text{r.n.}}, \end{cases} \quad (1)$$

where $\lfloor \bullet \rfloor$ is the floor function, $\psi(\bullet,\bullet)$, $c_{li}^{[0]}$, σ_{li}, and $\theta_{\text{r.n.}}$ are the receptive neuron's activation function, center, width and dead zone, respectively (r.n. in the last parameter means 'receptive neuron'), –1 indicates that the neuron does not fire.

In this work, we used Gaussian as activation function of receptive neurons:

$$\psi(|x_i(k) - c_{li}^{[0]}|, \sigma_{li}) = \exp\left(-\frac{(x_i(k) - c_{li}^{[0]})}{2\sigma_{li}^2}\right). \quad (2)$$

There can be several ways to set widths and centers of receptive neurons within a pool. As a rule, activation functions can be of two types – either 'narrow' or 'wide'. Centers of each width type of activation function are calculated in different ways but in either case they cover date range uniformly. More details can be found in [7].

Spiking neurons form the second hidden layer of the network. Spiking neuron is considered to be formed of two constituents, they are: synapse and soma. Synapses between receptive neurons and spiking neurons are multiple structures. A multiple synapse MS_{jli} consists of a set of q subsynapses with different time delays d^p, $d^p - d^{p-1} > 0$, $d^q - d^1 > t_{\max}^{[0]}$, and adjustable weights w_{jli}^p (here $p = \overline{1, q}$). It should be noted that number of subsynapses within a multiple synapse are fixed for the whole network. Having a spike $t_{li}^{[0]}(x_i(k))$ from the li-th receptive neuron, the p-th subsynapse of the j-th spiking neuron produces delayed weighted postsynaptic potential

$$u_{jli}^p(t) = w_{jli}^p \varepsilon_{jli}^p(t) = w_{jli}^p \varepsilon(t - (t_{li}^{[0]}(x_i(k)) + d^p)), \quad (3)$$

where $\varepsilon(\bullet)$ is a spike-response function usually described by the expression [13]

$$\varepsilon(t - (t_{li}^{[0]}(x_i(k)) + d^p)) = \frac{t - (t_{li}^{[0]}(x_i(k)) + d^p)}{\tau_{\text{PSP}}} \times$$
$$\times \exp\left(1 - \frac{t - (t_{li}^{[0]}(x_i(k)) + d^p)}{\tau_{\text{PSP}}}\right) \times$$
$$\times H(t - (t_{li}^{[0]}(x_i(k)) + d^p)), \quad (4)$$

τ_{PSP} – membrane potential decay time constant whose value can be obtained empirically (PSP means 'postsynaptic potential'), $H(\bullet)$ – the Heaviside step function. Output of the multiple synapse MS_{jli} forms total postsynaptic potential

$$u_{jli}(t) = \sum_{p=1}^{q} u_{jli}^p(t). \quad (5)$$

Each incoming total postsynaptic potential contributes to membrane potential of spiking neuron SN_j as follows:

$$u_j(t) = \sum_{i=1}^{n} \sum_{l=1}^{h} u_{jli}(t). \quad (6)$$

Spiking neuron SN_j generates at most one outgoing spike $t_j^{[1]}(x(k))$ during a simulation interval (the presentation of an input pattern $x(k)$), and fires at the instant the membrane potential reaches firing threshold $\theta_{\text{s.n.}}$

(s.n. means here ‘spiking neuron’). After neuron has fired, the membrane potential is reset to the rest value u_{rest} (0 usually) until the next input pattern is presented.

Number of spiking neurons in the second hidden layer is set to be equal to the number of clusters to be detected. Each spiking neuron corresponds to a certain cluster. In self-learning spiking neural network, the neuron that has fired first to the input pattern defines cluster that the pattern belongs to [7]. Firing time of spiking neuron $t_j^{[1]}(x(k))$ defines temporal distance of input pattern to center of the corresponding cluster.

The third layer, namely output fuzzy clustering layer, takes temporal distances of input pattern to centers of all classes and produces membership level according to fuzzy probabilistic approach [9]:

$$\mu_j(x(k)) = \frac{\left(t_j^{[1]}(x(k))\right)^{\frac{2}{1-\zeta}}}{\sum_{\iota=1}^{m}\left(t_\iota^{[1]}(x(k))\right)^{\frac{2}{1-\zeta}}}, \tag{7}$$

where m is the number of classes, $\zeta \geq 0$ is the fuzzifier that determines boundary between clusters and controls the amount of fuzziness in the final partition.

LEARNING ALGORITHM

The purpose of an unsupervised learning algorithm of spiking neural network is to adjust centers of spiking neurons so as to make each of them to correspond to centroid of a certain data cluster. Such learning algorithm was introduced on the basis of two learning rules, namely, ‘Winner-Takes-All’ rule and temporal Hebbain rule [13]. The first one defines which neuron should be updated, and the second one defines how it should be updated. The algorithm adjusts neuron centers through synaptic weights updating, whereas synaptic time delays always remain constant. The concept here is that significance of the given time delay can be changed by varying corresponding synaptic weight.

Each learning epoch consists of two phases. Competition, the first phase defines a neuron-winner. Being laterally linked with inhibitory connections, spiking neurons compete to respond to the pattern. The one wins (and fires) whose center is the closest to the pattern. After competition, weights adjusting takes place. The learning algorithm adjusts synaptic weights of the neuron-winner to move it closer to the input pattern. It strengthens weights of those subsynapses which contributed to the neuron-winner’s firing (i.e. the subsynapses produced delayed spikes right before the neuron firing) and weakens ones which did not contribute (i. e. the delayed spikes appeared right after the neurons firing or long before it). Generally, the learning algorithm can be expressed as

$$w_{jli}^p(K+1) = \begin{cases} w_{jli}^p(K) + \eta_w(K)L(\Delta t_{jli}^p), \ j = \tilde{j}; \\ w_{jli}^p(K), \ j \neq \tilde{j}, \end{cases} \tag{8}$$

where K is the current epoch number, $\eta_w(\bullet) > 0$ is the learning rate (while it is constant in [13], it can depended on epoch number in the general case; w means ‘weights’), $L(\bullet)$ is the learning function [7, 13], $\tilde{j}$ is the number of neuron-winner on the current epoch, Δt_{jli}^p is the time delay between delayed spike $t_{li}^{[0]}(x_i(k)) + d^p$ produced by the p-th subsynapse of the li-th synapse and spiking neuron firing time $t_j^{[1]}(x(k))$:

$$\Delta t_{jli}^p = t_{li}^{[0]}(x_i(k)) + d^p - t_j^{[1]}(x(k)). \tag{9}$$

As a rule, the learning function has the following form [9, 13]:

$$L(\Delta t_{jli}^p) = (1+\beta)\exp\left(-\frac{(\Delta t_{jli}^p - \alpha)^2}{2(\kappa-1)}\right) - \beta, \tag{10}$$

$$\kappa = 1 - \frac{\nu^2}{2\ln\left(\frac{\beta}{1+\beta}\right)}, \tag{11}$$

where $\alpha < 0$, $\beta > 0$, ν are the shape parameters of the learning function $L(\bullet)$ that can be obtained empirically [7, 13]. The learning function and its shape parameters are depicted on Fig. 1.

After learning stage, center of a spiking neuron represents centroid of a certain data cluster, and spiking neural network can successfully perform unsupervised classification of the input set.

The learning algorithm (8) updates only neuron-winner on each epoch and disregards other neurons. It seems more natural to update not only spiking neuron-

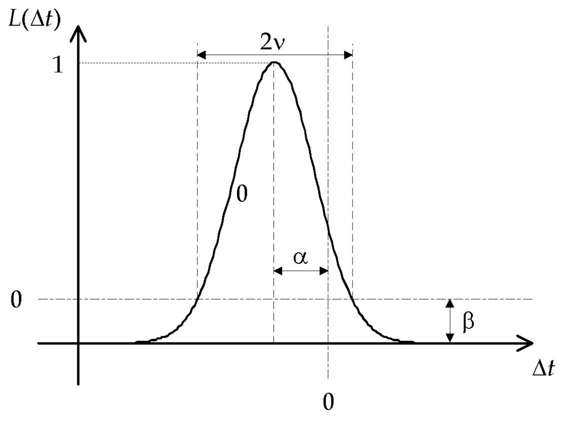

Fig. 1. Learning function $L(\bullet)$

winner, but also its neighbours. This approach is known as ‘Winner-Takes-More’ rule. It implies that there is a cooperation phase before weights adjustment. Neuron-winner determines a local region of topological neighbourhood on each learning epoch. Within this region, the neuron-winner fires along with its neighbours, and the closer a neighbour is to the winner, the more its weights are adjusted. The topological region is represented by the neighbourhood function $\varphi(|\Delta t_{\tilde{j}j}|)$ that depends on difference $|\Delta t_{\tilde{j}j}|$ between the neuron-winner firing time $t_{\tilde{j}}^{[1]}(x(k))$ and the neighbour firing time $t_{j}^{[1]}(x(k))$ (distance between the neurons in temporal sense) and a parameter that defines effective width of the region. As a rule, $\varphi(\bullet)$ is a kernel function that is symmetric about its maximum at the point where $\Delta t_{\tilde{j}j} = 0$. It reaches unity at that point and monotonically decreases as $\Delta t_{\tilde{j}j}$ tends to infinity. The functions that are the most frequently used as neighbourhood function are Gaussian, paraboloid, Mexican Hat, and many others [14].

For self-learning spiking neural network, the learning algorithm based on ‘Winner-Takes-More’ rule can be expressed in the following form:

$$w_{jli}^{p}(K+1) = w_{jli}^{p}(K) + \eta_{w}(K)\varphi(|\Delta t_{\tilde{j}j}|)L(\Delta t_{jli}^{p}), \quad (12)$$

where temporal distance $\Delta t_{\tilde{j}j}$ is

$$\Delta t_{\tilde{j}j} = t_{\tilde{j}}^{[1]}(x(k)) - t_{j}^{[1]}(x(k)). \quad (13)$$

Obviously, expression (12) is a generalization of (8).

Analysis of competitive unsupervised learning convergence showed that width parameter of the neighbourhood function should decrease during synaptic weights adjustment [15]. For Gaussian neighbourhood function

$$\varphi(|\Delta t_{\tilde{j}j}|, K) = \exp\left(-\frac{(\Delta t_{\tilde{j}j})^2}{2\rho^2(K)}\right) \quad (14)$$

width parameter ρ can be adjusted as follows [16]:

$$\rho(K) = \rho(0)\exp\left(\frac{K}{\gamma}\right), \quad (15)$$

where $\gamma > 0$ is a scalar that determines rate of neuron-winner effect on its neighbours.

Noteworthily that exponential decreasing of width parameter can be achieved by applying the simpler expression instead of (15) [14]:

$$\rho(K) = \gamma\rho(K-1), \ 0 < \gamma < 1. \quad (16)$$

Learning algorithm (12) requires modification of self-learning spiking neural architecture. Lateral inhibitory connections in the second hidden layer should be replaced with excitatory ones during the network learning stage in order to implement ‘Winner-Takes-More’ rule.

In the following sections, it will be shown that the learning algorithm based on ‘Winner-Takes-More’ rule is more natural than the one based on to ‘Winner-Takes-All’ rule to learn fuzzy spiking neural network.

FUZZY SPIKING NEURAL NETWORK AS ANALOG-DIGITAL SYSTEM

Hardware implementations of spiking neural network demonstrated fast processing ability that made it possible to apply such systems in real-life applications where processing speed was a rather critical parameter [5, 8]. From theoretical point of view, research works on spiking neurons hardware implementation subject are very particular, they lack for a technically plausible description on a general ground. In this section, we consider a spiking neuron as a processing system of classical automatic control theory. Spiking neuron functioning is described in terms of the Laplace transform.

Within a scope of automatic control theory, a spike $t(x(k))$ can be represented by the Dirac delta function $\delta(t - t(x(k)))$. Its Laplace transform is

$$L\{\delta(t - t(x(k)))\} = e^{-t(x(k))s}, \quad (17)$$

where s is the Laplace operator. Spiking neuron takes spikes on its input, performs spike – membrane potential – spike transformation, and produces spikes on its output. Obviously, it is a kind of analog-digital system that processes information in continuous-time form and transmits it in pulse-position form. This is the basic concept for designing analog-digital architecture of self-learning spiking neural network. Overall network architecture is depicted on Fig. 2.

Multiple synapse MS_{jli} of a spiking neuron SN_j transforms incoming pulse-position signal $\delta\left(t - t_{li}^{[0]}(x_i(k))\right)$ to continuous-time signal $u_{jli}(t)$. Spike-response function (4), the basis of such transformation, has form that is similar to the one of impulse response of second-order damped response unit. Transfer function of a second-order damped response unit with unit gain factor is

$$\tilde{G}(s) = \frac{1}{(\tau_1 s + 1)(\tau_2 s + 1)} = \frac{1}{\tau_4^2 s^2 + \tau_3 s + 1}, \quad (18)$$

where $\tau_{1,2} = \frac{\tau_3}{2} \pm \sqrt{\frac{\tau_3^2}{4} - \tau_4^2}$, $\tau_1 \geq \tau_2$, $\tau_3 \geq 2\tau_4$, and its impulse response is

$$\tilde{\varepsilon}(t) = \frac{1}{\tau_1 - \tau_2}\left(e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}}\right). \quad (19)$$

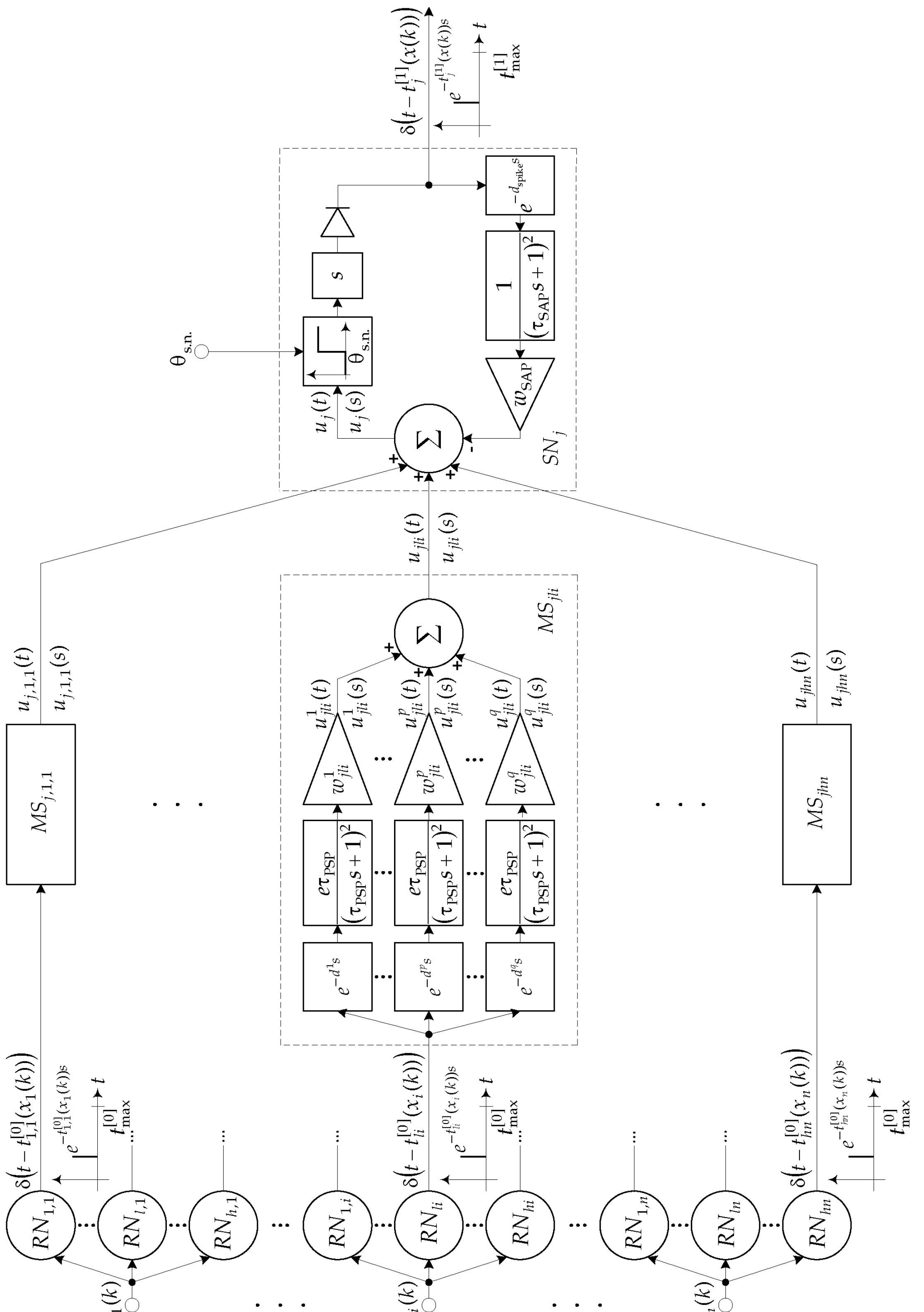

Fig. 2. Spiking neuron as a nonlinear dynamic system

Putting $\tau_1 = \tau_2 = \tau_{\mathrm{PSP}}$ (that corresponds to a second-order critically damped response unit) and applying l'Hôpital's rule, one can obtain

$$\tilde{\varepsilon}(t) = \frac{t}{\tau_{\mathrm{PSP}}^2} e^{-\frac{t}{\tau_{\mathrm{PSP}}}}. \tag{20}$$

Comparing spike-response function (4) with the impulse response (20) leads us to the following relationship:

$$\varepsilon(t) = \tau_{\mathrm{PSP}} e \tilde{\varepsilon}(t). \tag{21}$$

Thus, transfer function of the second-order critically damped response unit whose impulse response corresponds to a spike-response function is

$$G_{\mathrm{SRF}}(s) = \frac{e\tau_{\mathrm{PSP}}}{(\tau_{\mathrm{PSP}} s + 1)^2}, \tag{22}$$

where SRF means 'spike-response function'.

Now, we can design multiple synapse in terms of the Laplace transform. As illustrated on Fig. 2, multiple synapse MS_{jli} is a dynamic system that consists of a set of subsynapses that are connected in parallel. Each subsynapse is formed by a group of time delay, second-order critically damped response unit, and gain. As a response to incoming spike $\delta\left(t - t_{li}^{[0]}(x(k))\right)$, the subsynapse produces delayed weighted postsynaptic potential $u_{jli}^p(s)$, and the multiple synapse produces total postsynaptic potential $u_{jli}(s)$ that arrives to spiking neuron soma.

Taking into account (22), transfer function of the p-th subsynapse of MS_{jli} takes the following form:

$$U_{jli}^p(s) = w_{jli}^p e^{-d^p s} G_{\mathrm{SRF}}(s) = \frac{\tau_{\mathrm{PSP}} w_{jli}^p e^{1-d^p s}}{(\tau_{\mathrm{PSP}} s + 1)^2}, \tag{23}$$

and its response to a spike $\delta\left(t - t_{li}^{[0]}(x_i(k))\right)$ is

$$u_{jli}^p(s) = e^{-t_{li}^{[0]}(x_i(k))s} U_{jli}^p(s) = \frac{\tau_{\mathrm{PSP}} w_{jli}^p e^{1-(t_{li}^{[0]}(x_i(k)) + d^p)s}}{(\tau_{\mathrm{PSP}} s + 1)^2}. \tag{24}$$

So finally, considering transfer function of multiple synapse MS_{jli}

$$U_{jli}(s) = \sum_{p=1}^{q} U_{jli}^p(s) = \sum_{p=1}^{q} \frac{\tau_{\mathrm{PSP}} w_{jli}^p e^{1-d^p s}}{(\tau_{\mathrm{PSP}} s + 1)^2}, \tag{25}$$

the Laplace transform of the multiple synapse output can be expressed in the following form:

$$u_{jli}(s) = e^{-t_{li}^{[0]}(x_i(k))s} U_{jli}(s) = \\ = \sum_{p=1}^{q} \frac{\tau_{\mathrm{PSP}} w_{jli}^p e^{1-(t_{li}^{[0]}(x_i(k)) + d^p)s}}{(\tau_{\mathrm{PSP}} s + 1)^2}. \tag{26}$$

Spiking neuron soma performs transformation that is opposite to one of the synapse. It takes continuous-time signals $u_{jli}^p(t)$ and produces pulse-position signal $\delta\left(t - t_j^{[1]}(x(k))\right)$. In doing so soma responds each time membrane potential reaches a certain threshold value. In other words, spiking neuron soma acts as a threshold detection system and consequently it can be designed on the base of bang-bang control systems concept [17].

Threshold detection behaviour of a neuron soma can be modelled by an element relay with dead zone $\theta_{\mathrm{s.n.}}$ that is defined by the nonlinear function

$$\Phi_{\mathrm{relay}}(u_j(t), \theta_{\mathrm{s.n.}}) = \frac{\mathrm{sign}(u_j(t) - \theta_{\mathrm{s.n.}}) + 1}{2}, \tag{27}$$

where sign(•) is the signum function. Soma firing can be described by a derivative unit that is connected with the element relay in series and produces a spike each time the relay switches. In order to avoid a negative spike that appears as a response to the relay resetting, a usual diode is added next to the derivative unit. The diode is defined by the following function:

$$\Phi_{\mathrm{diode}}(\delta(t - t_{\mathrm{relay}}^{[1]})) = \delta(t - t_{\mathrm{realy}}^{[1]}) H(\delta(t - t_{\mathrm{relay}}^{[1]})), \tag{28}$$

where $t_{\mathrm{relay}}^{[1]}$ is a spike produced by the derivative unit upon the relay switching.

Now we can define the Laplace transform of an outgoing spike $t_j^{[1]}(x(k))$, namely,

$$L\left\{\delta\left(t - t_j^{[1]}(x(k))\right)\right\} = e^{-t_j^{[1]}(x(k))s} = \\ = L\{\Phi_{\mathrm{diode}}(sL\{\Phi_{\mathrm{relay}}(u_j(t), \theta_{\mathrm{s.n.}})\})\}. \tag{29}$$

As it was mentioned above, the leaky integrate-and-fire model disregards the neuron refractoriness. Anyway, the refractory period is implemented in the layer of spiking neurons indirectly. The point is that a spiking neuron cannot produce another spike after firing and until the end of the simulation interval since the input pattern is provided only once within the interval. In the analog-digital architecture of spiking neuron, the refractoriness can be modelled by a feedback circuit. As shown on Fig. 2, it is a group of a time delay, a second-order critically damped response unit, and a gain that are connected in series. The time delay defines duration of a spike generation period d_{spike} (usually, $d_{\mathrm{spike}} \to 0$). The second-order critically damped response unit defines a spike after-potential. Generally, spike after-potential can be represented by a second-order damped response unit, but for the sake of simplicity, we use critically damped response unit as it can be defined by one parameter only, namely, τ_{SAP} (SAP means here 'spike after-potential').

This parameter controls duration of the refractory period. Finally, the gain unit sets amplitude of the spike after-potential w_{SAP}. Obviously, w_{SAP} should be much greater than any synaptic weight.

Thus, transfer function of the feedback circuit is

$$G_{\mathrm{F.B.}}(s) = \frac{w_{\mathrm{SAP}} e^{-d_{\mathrm{spike}} s}}{(\tau_{\mathrm{SAP}} s + 1)^2}, \qquad (30)$$

where F.B. means ‘feedback circuit’, and transfer function of the soma is

$$G_{\mathrm{soma}}(s) = \frac{G_{\mathrm{F.F.}}}{1 + G_{\mathrm{F.F.}} G_{\mathrm{F.B.}}}, \qquad (31)$$

where $G_{\mathrm{F.F.}}$ is defined by (29) (F.F. means ‘feed forward circuit’).

It is easily seen that the functioning of spiking neuron analog-digital architecture introduced above is similar to the spike-response model [6].

CONCLUSIONS

Spiking neural networks are more realistic models of real neuronal systems than artificial neural networks of the previous generations. Nevertheless, they can be describedin a strict technically plausible way based on the Laplace transform. Spiking neural network designed in terms of transfer functions is an analog-digital nonlinear dynamic system that conveys and processes information both in pulse-position and continuous-time forms. Such precise formal description of spiking neural network architecture and functioning provides researchers and engineers with a framework to construct hardware implementations of various spiking neural networks for real-time data processing of different levels of complexity.

REFERENCES

1. *Haykin S.* Neural Networks: A Comprehensive Foundation / Haykin S. – Upper Saddle River : Prentice Hall, 1999. – 842 p.
2. *Bezdek J. C.* Fuzzy Models and Algorithms for Pattern Recognition and Image Processing / Bezdek J. C., Keller J., Krishnapuram R., Pal N. R. – New York : Springer, 2005. – 776 p.
3. *Sato-Ilic M.* Innovations in Fuzzy Clustering / Sato-Ilic M., Jain L. C. – Berlin – Heidelberg – New York : Springer, 2006. – 152 p.
4. *Jang J.-Sh. R.* Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence / Jang J.-Sh. R., Sun Ch.-T., Mizutani E. – Upper Saddle River : Prentice Hall, 1997. – 614 p.
5. *Maass W.* Networks of spiking neurons: the third generation of neural network models / Maass W. // Neural Networks. – 1997. – 10. – P. 1659–1671.
6. *Gerstner W.* Spiking Neuron Models / Gerstner W., Kistler W. M. – Cambridge : The Cambridge University Press, 2002. – 400 p.
7. *Bohte S. M.* Unsupervised clustering with spiking neurons by sparse temporal coding and multi-layer RBF networks / Bohte S. M., Kok J. S., La Poutre H. // IEEE Trans. on Neural Networks. – 2002. – 13. – P. 426–435.
8. *Maass W.* Pulsed Neural Networks / Maass W., Bishop C. M. – Cambridge, MA : The MIT Press, 1998. – 408 p.
9. *Bodyanskiy Ye.* A self-learning spiking neural network for fuzzy clustering task / Bodyanskiy Ye., Dolotov A. // Sci. Proc. of Riga Technical University: Information Technology and Management Science. – 2008. – 36. – P. 27–33.
10. *Bodyanskiy Ye.* Fuzzy possibilistic clustering using self-learning spiking neural network / Bodyanskiy Ye., Dolotov A., Pliss I., Viktorov Ye. // Wissenschaftliche Berichte der Hochschule Zittau/Goerlitz. – 2008. – Heft 100. – Nr. 2360–2395. – S. 53–60.
11. *Bodyanskiy Ye.* Fuzzy spiking neural network and design of experiments alphabetical optimality criteria using in data clustering problem / Bodyanskiy Ye., Dolotov A. // Proc. Int. Conf. “Intellectual Systems for Decision Making and Problems of Computational Intelligence (ISDMCI’2009)” (May 18–22, 2009, Yevpatoria, Ukraine). – Kherson : Kherson National Technical University, 2009. – Vol. 2. – P. 154–157.
12. *Bodyanskiy Ye.* Adaptive Gustafson-Kessel fuzzy clustering algorithm based on self-learning spiking neural network / Bodyanskiy Ye., Dolotov A., Pliss I. ; eds. by G. Setlak, K. Markov. – Rzeszow : Rzeszow University of Technology, 2009. – P. 17–24. – (International Book Series “Information Science and Computing” ; No. 13).
13. *Natschlaeger T.* Spatial and temporal pattern analysis via spiking neurons / Natschlaeger T., Ruf B. // Network: Computations in Neural Systems. – 1998. – 9. – P. 319–332.
14. *Бодянский Е. В.* Искусственные нейронные сети: архитектуры, обучение, применения / Бодянский Е. В., Руденко О. Г. – Харьков : ТЕЛЕТЕХ, 2004. – 372 с.
15. *Cottrel M.* A stochastic model of retinotopy: a self-organizing process / Cottrel M., Fort J. // Biological Cybernetics. – 1986. – 53. – P. 405–411.
16. *Ritter H.* On stationary state of the Kohonen self-organizing sensory mapping / Ritter H., Schulten K. // Biological Cybernetics. – 1986. – 54. – P. 239–249.
17. *Цыпкин Я. З.* Теория релейных систем автоматического управления / Цыпкин Я. З. – М. : Государственное издательство технико-теоретической литературы, 1955. – 456 с.

Надійшла 24.08.2009

Бодянський Є., Долотов А.

АНАЛОГОВО-ЦИФРОВА САМОНАВЧАЛЬНА ФАЗЗІ-СПАЙК-НЕЙРОННА МЕРЕЖА ТА АЛГОРИТМ ЇЇ НАВЧАННЯ НА ОСНОВІ ПРАВИЛА «ПЕРЕМОЖЕЦЬ ОТРИМУЄ БІЛЬШЕ»

У цій роботі запропоновано аналогово-цифрову архітектуру фаззі-спайк-нейронної мережі, що самонавчається. Синапс та сому спайк-нейрона розглянуто з позиції класичної теорії автоматичного керування. Застосуванням правила «переможець отримує більше» поліпшено стандартний алгоритм самонавчання спайк-нейронної мережі.

Ключові слова: аналогово-цифрова архітектура, самонавчальна фаззі-спайк-нейронна мережа, теорія автоматичного керування, алгоритм навчання без учителя, правило «переможець отримує більше».

Бодянский Е., Долотов А.

АНАЛОГОВО-ЦИФРОВАЯ САМООБУЧАЮЩАЯСЯ ФАЗЗИ-СПАЙК-НЕЙРОННАЯ СЕТЬ И АЛГОРИТМ ЕЕ ОБУЧЕНИЯ НА ОСНОВЕ ПРАВИЛА «ПОБЕДИТЕЛЬ ПОЛУЧАЕТ БОЛЬШЕ»

В настоящей работе предложена архитектура аналогово-цифровой самообучающейся фаззи-спайк-нейронной сети. Синапс и сома спайк-нейрона рассмотрены с позиции теории автоматического управления. Посредством применения правила «победитель получает больше» усовершенствован стандартный алгоритм самообучения спайк-нейронной сети.

Ключевые слова: аналогово-цифровая архитектура, самообучающаяся фаззи-спайк-нейронная сеть, теория автоматического управления, алгоритм обучения без учителя, правило «победитель получает больше».

УДК 681.51: 622.7

Купін А. І.

Канд. техн. наук, завідувач кафедри Криворізького технічного університету

СТРУКТУРА ПРОТОТИПУ ТА ОБҐРУНТУВАННЯ ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ ЗБАГАЧЕННЯ ЗАЛІЗНОЇ РУДИ

Наведено структурну схему прототипу інтелектуальної системи керування секцією магнітного збагачення залізної руди. На основі результатів промислових випробувань зроблена оцінка економічної ефективності впровадження подібної системи в умовах збагачувального виробництва.

Ключові слова: інтелектуальна система керування, збагачення, залізна руда, ефективність.

ВСТУП

У наш час достатньо актуальною проблемою для вітчизняних підприємств гірничо-металургійної галузі промисловості є підвищення конкурентоспроможності виробництва за рахунок зменшення собівартості переділу, оптимізації енерговитрат, стабілізації або поліпшення якості продукції тощо. Загальновідомо, що одним з найбільш перспективних шляхів вирішення цієї проблеми є комплексна автоматизація технологічних процесів (ТП) [1, 2].

Технологічні процеси збагачення руд чорних металів (магнетитових кварцитів) є достатньо складними об’єктами автоматизації. Це обумовлено їх багатовимірністю та багатостадійністю, властивостями нелінійності та нестаціонарності, значним запізненням інформаційних показників у часі, наявністю нечіткої та неповної інформації. У зв’язку з цим можливості застосування класичних підходів теорії автоматизованого керування є доволі обмеженими [3].

Теоретичні дослідження, комп’ютерне моделювання та промислові випробування [3–9] довели потенціальні можливості застосування інтелектуальних підходів щодо ідентифікації, керування та оптимізації ТП збагачення магнетитових кварцитів. У першу чергу це стосується сучасних напрямів розвитку штучного інтелекту: нейрокерування, нечіткої логіки, класифікаційного керування та еволюційної оптимізації.

Враховуючи, що поточний стан дослідження можливостей застосування подібних технологій у гірничій справі поки містить доволі обмежену кількість реальних впроваджень на вітчизняних підприємствах, постійно виникають питання щодо техніко-економічного обґрунтування таких розробок. Тому метою статті є оцінка ефективності впровадження інтелектуальної системи керування в умовах технологічної лінії (секції) рудозбагачувальної фабрики (РЗФ) гірничо-збагачувального комбінату (ГЗК).

ПОСТАНОВКА ЗАВДАННЯ

Згідно з рішеннями, розглянутими у попередніх роботах автора [3, 5–8], розглядається така схема реалізації ІСК (рис. 1).

Основою системи є апаратно-програмне ядро ІСК (блок № 1), що складається з п’яти підсистем.

1.1. Інтерфейсна частина, сервер та монітор SCADA. Реалізується на підставі спеціалізованого програмного забезпечення типу: Контур, Monitor Pro, Scantic, Trace Mode [10–14]. Ця підсистема виконує функції візуалізації (моніторингу) ходу ТП, введення та контролювання уставок технологічних параметрів, формування звітів. На апаратному рівні такі підсистеми реалізуються на підставі застосування архітектури «клієнт – сервер». В якості серверів робочих станцій застосовується переважно комп’ютерне обладнання у промисловому виконанні з підвищеним рівнем надійності (стандарти IP50, IP65-67 [13]). Клієнтські станції реалізуються у вигляді автоматизованих робочих місць (АРМ) спеціалістів (наприклад, технолога, диспетчера РЗФ). Для інфор-